AMENDMENTS TO THE CLAIMS

Applicant submits below a complete listing of the current claims, including marked-up claims with insertions indicated by underlining and deletions indicated by strikeouts and/or double bracketing. This listing of claims replaces all prior versions, and listings, of claims in the application:

1-6. (Canceled)

7. (Currently Amended) An object model embodied on a computer-readable medium for managing a service on a computer, the object model comprising:

a policy object model for specifying, by a first user, ~~used to specify~~ one or more policies that the service supports in a packet-centric form, and, by a second user, said one or more policies in a user-centric form and/or an application-centric form; and

a policy engine platform for interacting of said first user with said one or more policies specified in said packet-centric form ~~for the service~~ and of said second user with said one or more policies specified in said user-centric form and/or said application-centric form ~~at least one component that actually performs the service~~, and to provide said one or more policies to said at least one component that actually performs the service.

8. (Currently Amended) The object model of claim 7, wherein the policy engine platform comprises a rule editor for adding an additional policy by said first user in accordance with the policy object model.

9. (Currently Amended) The object model of claim 8, wherein the rule editor is also configured by said first user to delete a policy.

10. (Currently Amended) The object model of claim 8, wherein the rule editor is also configured by said first user to edit a policy.

11. (Original) The object model of claim 7, wherein the policy engine platform comprises a setting editor configured to automatically generate a policy based upon an application and user combination.

12. (Currently Amended) The object model of claim 11, wherein the setting editor generates a plurality of policies, and is further configured to permit [[a]] said second user to select from the plurality of policies.

13. (Currently Amended) The object model of claim 12, wherein the setting editor is further configured by said second user to permit setting one of the plurality of policies as a default policy.

14. (Original) The object model of claim 7, wherein the policy engine platform comprises a rule explorer for providing a view of the one or more policies.

15. (Original) The object model of claim 7, wherein the policy object model comprises a policyrule object usable to generate policy, the policyrule object comprising a condition property and an action property, wherein a policy generated by the policyrule object is configured to perform an action in the action property responsive to a condition in the condition property being met.

16. (Original) The object model of claim 7, wherein the service is a firewall service.

17. (Original) The object model of claim 7, wherein the policy engine platform is configured to deny providing said one or more policies to the component if a requestor is not authorized.

18. (Original) The object model of claim 17, wherein determining whether a requestor is authorized comprises comparing a provider rank for the requestor against a permitted rank, and if

the provider rank for the requestor does not meet or exceed the permitted rank, denying the requestor.

19. (Currently Amended) A method of managing a service on a computer, the method comprising:

specifying, via a policy object model, by a first user, one or more policies that the service supports in a packet-centric form, and, by a second user, said one or more policies in a user-centric form and/or an application-centric form; ~~and~~

interacting, via a policy engine platform, of said first user with said one or more policies ~~for the service~~ specified in said packet-centric form, and of said second user with said one or more policies specified in said user-centric form and/or said application-centric form ~~at least one component that actually performs the service~~; and

providing, via the policy engine platform, said one or more policies to ~~said~~ at least one component that actually performs the service.

20. (Original) The method of claim 19, further comprising automatically generating a policy based upon an application and user combination.

21. (Currently Amended) The method of claim 20, further comprising generates a plurality of policies, and permitting a user to select from the plurality of policies.

22. (Currently Amended) The method of claim 21, further comprising setting one of the plurality of policies as a default policy.

23. (Currently Amended) The method of claim 22, further comprising authorizing a user prior to providing said one or more policies.

24. (Currently Amended) An object model embodied on a computer-readable medium for managing a firewall service on a computer, the object model comprising a policy object model

used to specify, by a first user, one or more policies that the firewall service supports in a packet-centric form, and, by a second user, said one or more policies in a user-centric form and/or an application-centric form, the policy model comprising a policyrule object usable to generate policy, the policyrule object comprising a condition property and an action property, wherein a policy generated by the policyrule object is configured to perform an action in the action property responsive to a condition in the condition property being met.

25. (Original) The object model of claim 24, further comprising an IPSecRule derived from the policyrule object, the IPSecRule being configured to trigger an IPSec callout when an IPSec condition is matched, and to indicate configuration parameters for securing traffic related to the callout.

26. (Original) The object model of claim 25, wherein the IPSecRule evaluates a standard 5-tuple to determine if a condition has been met.

27. (Original) The object model of claim 24, further comprising a KeyingModuleRule derived from the policyrule object, the KeyingModuleRule being configured to select which key negotiation module to use when there is no existing secure channel to a remote peer.

28. (Original) The object model of claim 27, wherein the KeyingModuleRule evaluates a standard 5-tuple to determine if a condition has been met.

29. (Original) The object model of claim 24, further comprising a IKERule derived from the policyrule object and configured to specify the parameters for carrying out Internet Key Exchange key negotiation protocol.

30. (Original) The object model of claim 29, wherein the IKERule evaluates a local address and a remote address to determine if a condition has been met.